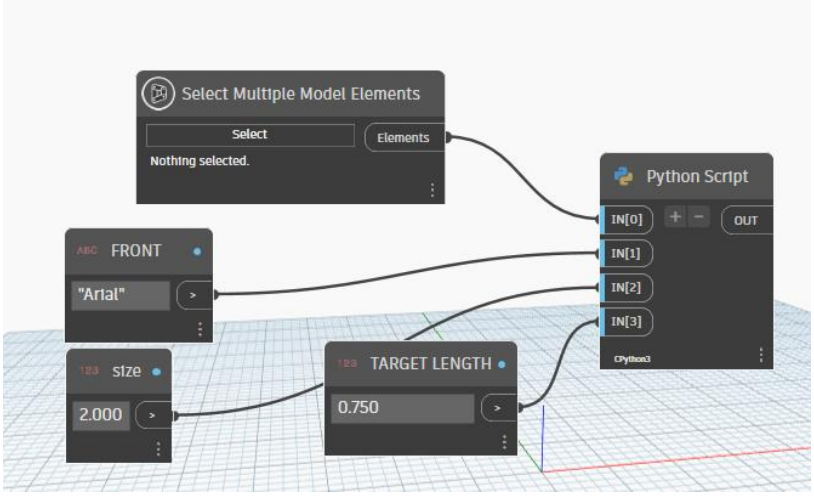
SCHEDULE RESIZER (P9.1)

```
import clr
clr.AddReference('System.Drawing')
clr.AddReference('RevitAPI')
clr.AddReference('RevitServices')

from System.Drawing import Graphics, Font, GraphicsUnit, Bitmap
from Autodesk.Revit.DB import *
from RevitServices.Persistence import DocumentManager
from RevitServices.Transactions import TransactionManager

doc = DocumentManager.Instance.CurrentDBDocument

# --- INPUTS FROM DYNAMO ---
# Forces the input to always be treated as a list, even if it's just 1 schedule
input_raw = IN[0] if isinstance(IN[0], list) else [IN[0]]
schedule_instances = UnwrapElement(input_raw)

font_name = str(IN[1])
font_size_mm = float(IN[2])
target_total_ft = float(IN[3])

# Setup Text Measuring Engine
g = Graphics.FromImage(Bitmap(1, 1))
g.PageUnit = GraphicsUnit.Point
text_font = Font(font_name, font_size_mm * 2.83465)

# Start the transaction ONCE for all schedules to maximize speed
TransactionManager.Instance.EnsureInTransaction(doc)

output_messages = []

# --- THE MULTI-SELECTION LOOP ---
for sched_inst in schedule_instances:
    if isinstance(sched_inst, ScheduleSheetInstance):
        try:
            view_sched = doc.GetElement(sched_inst.ScheduleId)
            definition = view_sched.Definition
            visible_fields =
[definition.GetField(f_id) for f_id in definition.GetFieldOrder() if not definition.GetField(f_id).IsHidden]
            section_data = view_sched.GetTableData().GetSectionData(SectionType.Body)
```

```
        # ----------------------------------------------------------------
        # STEP 1 & 2: UNWRAP ALL & MAKE FULLY TIGHT BASED ON LARGEST TEXT
        # ----------------------------------------------------------------
        natural_widths_ft = []

        for c in range(len(visible_fields)):
            # Step 1: Force wrapping OFF for everything initially
            visible_fields[c].IsWrapped = False

            # Step 2: Find the absolute largest text footprint in this column
            # Check Header Text size
            max_pt = g.MeasureString(str(visible_fields[c].ColumnHeading), text_font).Width

            # Check Body Rows text size
            for r in range(section_data.NumberOfRows):
                cell_val = str(view_sched.GetCellText(SectionType.Body, r, c))
                w = g.MeasureString(cell_val, text_font).Width
                if w > max_pt:
                    max_pt = w

            # Keep your customized point conversion factor
            natural_widths_ft.append((max_pt / 550.0) * 1)

        # ----------------------------------------------------------------
        # STEP 3: CHECK TOTAL LENGTH VS TARGET LENGTH
        # ----------------------------------------------------------------
        current_total_ft = sum(natural_widths_ft)

        # ----------------------------------------------------------------
        # STEP 4: IF LESS -> GIVE MORE PADDING IN EQUAL RATIO
        # ----------------------------------------------------------------
        if current_total_ft <= target_total_ft:
            extra_space = target_total_ft - current_total_ft
            space_per_col = extra_space / len(visible_fields)

            for i in range(len(visible_fields)):
                visible_fields[i].IsWrapped = False # Keep single-line text
                visible_fields[i].GridColumnWidth = natural_widths_ft[i] + space_per_col

            output_messages.append("Success: '{}' was narrow. Stretched columns
evenly.".format(view_sched.Name))

        # ----------------------------------------------------------------
        # STEP 5: IF GREATER -> START SQUEEZING FROM LARGEST COLUMN
```

```
            # ----------------------------------------------------------------
            else:
                largest_col_idx = natural_widths_ft.index(max(natural_widths_ft))
                total_minor_width = sum(natural_widths_ft) - natural_widths_ft[largest_col_idx]
                available_for_large_col = target_total_ft - total_minor_width

                for i in range(len(visible_fields)):
                    if i == largest_col_idx:
                        visible_fields[i].IsWrapped = True
                        visible_fields[i].GridColumnWidth = max(available_for_large_col, 0.20)
                    else:
                        visible_fields[i].IsWrapped = False
                        visible_fields[i].GridColumnWidth = natural_widths_ft[i]

                output_messages.append("Success: '{}' exceeded limit. Squeezed column index
{}.".format(view_sched.Name, largest_col_idx))

        except Exception as e:
            output_messages.append("Error processing schedule: {}".format(str(e)))
    else:
        output_messages.append("Skipped: Element is not a ScheduleSheetInstance.")

# Force Revit to refresh the screen graphics once for the whole batch
doc.Regenerate()
TransactionManager.Instance.TransactionTaskDone()

OUT = output_messages
```